Anonymous HIBE from Standard Assumptions over Type-3 Pairings using Dual System Encryption

Somindu C. Ramanna Pal

Palash Sarkar

Applied Statistics Unit Indian Statistical Institute 203, B.T. Road, Kolkata India 700108. e-mail: {somindu_r,palash}@isical.ac.in

Abstract

We present the first anonymous hierarchical identity based encryption (HIBE) scheme using Type-3 pairings with adaptive security based on standard assumptions. Previous constructions of anonymous HIBE schemes did not simultaneously achieve all these features. The new construction uses dual pairing vector spaces using an identity hash earlier used by Boneh, Boyen and Goh. The proof of security follows dual system approach based on decisional subspace assumptions which are implied by Symmetric eXternal Diffie-Hellman (SXDH) assumption in Type-3 pairing groups.

Keywords: hierarchical identity-based encryption (HIBE), anonymous HIBE, asymmetric pairings, dual-system encryption

1 Introduction

Identity-based encryption (IBE) allows a sender to encrypt a message using a receiver's identity itself as the public key. The receiver decrypts using its key obtained securely from a trusted authority called private key generator (PKG) that distributes keys to all users. Hierarchical IBE (HIBE) enables PKG to delegate key generation abilities to 'lower-level' entities, thus reducing its own computation overhead. A well-studied interesting property of (H)IBE systems is *anonymity* which requires that a ciphertext does not reveal the identity of the intended recipient. Anonymous HIBE was first formalised and studied by Abdalla *et.al* [ABC⁺05]. These schemes also lead to constructions of protocols such as public key encryption with keyword search (PEKS), public key encryption with temporary keyword search (PETKS) and identitybased encryption with keyword search (IBEKS). These protocols have one common goal - to enable search (based on some keywords) on encrypted documents along with the capability of delegating search.

Most known constructions of anonymous HIBE schemes are based on pairings. A pairing is a bilinear map $e: \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that is efficiently computable and non-degenerate. It is defined over groups \mathbb{G}_1 , \mathbb{G}_2 , \mathbb{G}_T all having the same order N. N could be either composite or prime; also \mathbb{G}_1 and \mathbb{G}_2 could be the same group. Prime-order pairings where the source groups (i.e., \mathbb{G}_1 and \mathbb{G}_2) are different (known as asymmetric pairings) have the most compact and fast implementations, especially if there are no known isomorphism between \mathbb{G}_1 and \mathbb{G}_2 that can be computed efficiently. Such pairings are called Type-3 pairings.

The focus of this work is anonymous HIBE schemes that can be instantiated using prime-order asymmetric pairings. Usually, achieving anonymity is considered to be more tricky compared to non-anonymous

Scheme	Pairing	Security	Assumptions	Degradation
[BW06]	Type-1	Selective-id	DLin, DBDH	O(1)
[SKOS09]	Composite	Selective-id	h-weak BDHI*, h -composite DH	<i>O</i> (1)
[DCIP10]	Composite	Adaptive-id	Decisional subgroup	O(q)
[Duc10]	Type-3	Selective-id	DBDH, \mathcal{P} -BDH	O(1)
[PL13]	Type-3	Selective-id	h-DBDHE, Augmented h -DLin	O(1)
[LPL13]	Type-3	Adaptive-id	LW1, LW2, DBDH, SXDH, Asymmetric 3DH	O(q)
[RS12]	Type-3	Adaptive-id	LW1, LW2, DBDH, A1	O(q)
This work	Type-3	Adaptive-id	SXDH	O(q)

Table 1: Comparison of anonymous HIBE schemes based on pairings without random oracles.

HIBE. Several constructions have been proposed prior to this work. The first construction without random oracles was given by Boyen and Waters [BW06] with a proof of security in the (weak) selective identity model. Later constructions in [SKOS09, DCIP10] achieve the stronger notion of adaptive-identity security but are instantiated with composite-order pairings. which are very inefficient compared to prime-order pairings. On the other hand, composite order pairing has richer structure that is helpful in applying dual system encryption [Wat09] techniques to prove adaptive security. On the other hand, prime-order pairings do not have this structure. Schemes based on prime-order asymmetric pairings are also known [Duc10, PL13]. Both these schemes are only selectively secure. The works [LPL13, RS12] obtain adaptively secure anonymous HIBE using Type-3 pairings but a drawback is that security is based on some non-standard (but static) assumptions.

Our Contribution. Our contribution is an anonymous HIBE scheme over Type-3 pairings with adaptive security based on standard computational assumptions. The construction uses *dual pairing vector spaces* introduced in [OT08, OT09]. These are mathematical structures that can be built upon Type-3 pairings which are useful in achieving anonymity and at the same time enjoy the "nice" properties that are found in composite order pairings suitable for dual system proofs. We combine the Boneh-Boyen-Goh [BBG05] (BBG) technique to hash the identity and Boyen-Waters [BW06] method to enable rerandomisation during key delegation. Security is proved using the dual system technique [Wat09, LW10], where two kinds of ciphertexts and keys are defined - *normal* and *semi-functional* that play a major role in arguing about adaptive security. The main challenge we faced was in defining the keys, the *semi-functional space* and creating sufficient amount of randomness to generate them during simulation. We get around this problem by working over vector spaces of dimension linear in the maximum depth of the HIBE.

Table 1 provides a comparison of our scheme with existing anonymous HIBE schemes. Here h denotes the maximum depth of the HIBE and q is the total number of key extraction queries in the security game. Note that in contrast to all other schemes, our scheme achieves adaptive security based on SXDH assumption which is a standard assumption in Type-3 pairings. As a result, our scheme is the first instance of fully secure anonymous HIBE based on standard assumptions over Type-3 pairings.

2 Preliminaries

Here we provide a few notation and preliminary definitions. The definition of HIBE and its security are provided in Appendix A.

2.1 Notation

For a set \mathcal{X} , the notation $x_1, \ldots, x_k \in_{\mathbb{R}} \mathcal{X}$ (or $x_1, \ldots, x_k \xleftarrow{\mathbb{R}} \mathcal{X}$) indicates that x_1, \ldots, x_k are elements of \mathcal{X} chosen independently at random according to some distribution R. We use the two notations interchangeably. The uniform distribution is denoted by U. For a (probabilistic) algorithm $\mathcal{A}, x \leftarrow \mathcal{A}(\cdot)$ means that x is chosen according to the output distribution of \mathcal{A} (which of course may be determined by its input). For two integers a < b, the notation [a, b] represents the set $\{x \in \mathbb{Z} : a \le x \le b\}$. Let \mathbb{G} be a finite cyclic group and \mathbb{G}^{\times} denote the set of generators of \mathbb{G} .

For a prime p, we denote by \mathbb{Z}_p the field of order p and \mathbb{Z}_p^n the vector space of all n-tuples over \mathbb{Z}_p . Vectors over \mathbb{Z}_p will be represented by lower case letters with an arrow on top (e.g. \vec{v}). $\vec{0}$ is the all-zero vector. For vectors $\vec{u} = (u_1, \ldots, u_n)^T$ and $\vec{v} = (v_1, \ldots, v_n)^T$, $\langle \langle \vec{u}, \vec{v} \rangle$ denotes their inner product $\sum_{i=1}^n u_i v_i$. $\mathbb{Z}_p^{n \times n}$ denotes the set of all $n \times n$ matrices over \mathbb{Z}_p and $GL(n, \mathbb{Z}_p)$, the general linear group of degree n over \mathbb{Z}_p . Matrices over \mathbb{Z}_p are represented by bold face upper case letters (e.g. \mathbf{X}). Let \mathbb{V} and \mathbb{V}^* be vector spaces over \mathbb{Z}_p obtained by direct product of groups \mathbb{G}_1 and \mathbb{G}_2 of order p. Vectors in \mathbb{V} shall be expressed as bold face letters, for example \mathbf{b} . The asterisk symbol is used to differentiate between vectors of \mathbb{V} and \mathbb{V}^* with the same names (e.g. \mathbf{b} and \mathbf{b}^*). We denote matrices over \mathbb{Z}_p by bold face upper case letters (e.g. \mathbf{X}) and matrices over \mathbb{G}_1 (resp. \mathbb{G}_2) by blackboard bold font, e.g. \mathbb{A} (resp. \mathbb{A}^*). \mathbb{A}^T denotes the transpose of matrix \mathbb{A} .

2.2 Pairings

Definition 2.1. Let \mathbb{G}_1 , \mathbb{G}_2 and \mathbb{G}_T be cyclic groups of prime order p. $\mathbb{G}_1, \mathbb{G}_2$ are additively written and \mathbb{G}_T is multiplicatively written. A pairing is a tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ where $\mathbb{G}_1 = \langle P_1 \rangle$, $\mathbb{G}_2 = \langle P_2 \rangle$ and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable non-degenerate bilinear map i.e., for elements $Q_1 \in \mathbb{G}_1$ and $Q_2 \in \mathbb{G}_2$, $e(Q_1, Q_2) = 1$ iff $Q_1 = 0$ or $Q_2 = 0$ and for $x, y \in \mathbb{Z}_p$, $e(xP_1, yP_2) = e(P_1, P_2)^{xy}$.

A bilinear map is called symmetric or a Type-1 bilinear map if $\mathbb{G}_1 = \mathbb{G}_2$; otherwise it is asymmetric. Asymmetric bilinear maps are further classified into Type-2 and Type-3 bilinear maps. In the Type-2 setting, there is an efficiently computable isomorphism either from \mathbb{G}_1 to \mathbb{G}_2 or from \mathbb{G}_2 to \mathbb{G}_1 whereas in the Type-3 setting there are no such isomorphisms known. We will denote elements of \mathbb{G}_1 by upper case letters with subscript 1 and elements of \mathbb{G}_2 with subscript 2.

2.3 Review of Dual Pairing Vector Spaces

Dual Pairing Vector Space from Asymmetric Pairings: A typical construction for a DPVS $\mathcal{V}_n = (p, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \bar{e}, \mathbb{A}, \mathbb{A}^*)$ is via a direct product over a pairing $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$. We consider such constructions over asymmetric pairings, in particular, Type-3 pairings because they have more efficient implementations both in terms of representation of group elements and pairing computation. Details of the construction are provided below.

- $\mathbb{V} = \underbrace{\mathbb{G}_1 \times \cdots \times \mathbb{G}_1}_{n \text{ times}}$ and $\mathbb{V}^* = \underbrace{\mathbb{G}_2 \times \cdots \times \mathbb{G}_2}_{n \text{ times}}$ are vector spaces of dimension n over \mathbb{Z}_p . Their vectors can be expressed as $\mathbf{x} = (X_1, \dots, X_n)$ and $\mathbf{y} = (Y_1, \dots, Y_n)$ respectively, where $X_j \in \mathbb{G}_1$ and $Y_j \in \mathbb{G}_2$.
- \mathbb{A}, \mathbb{A}^* form the canonical bases of \mathbb{V}, \mathbb{V}^* respectively i.e., $\mathbb{A} = (\mathbf{a}_1, \dots, \mathbf{a}_n)$ and $\mathbb{A}^* = (\mathbf{a}_1^*, \dots, \mathbf{a}_n^*)$ with $\mathbf{a}_i = (\underbrace{0, \dots, 0}_{i-1}, P_1, \underbrace{0, \dots, 0}_{n-i})^T$ and $\mathbf{a}_i^* = (\underbrace{0, \dots, 0}_{i-1}, P_2, \underbrace{0, \dots, 0}_{n-i})^T$ for $i = 1, \dots, n$.

• Let $\mathbf{x} = (\vec{x})_{\mathbb{A}}$ and $\mathbf{y} = (\vec{y})_{\mathbb{A}^*}$. The function \bar{e} is defined as

$$\bar{e}(\mathbf{x}, \mathbf{y}) = \prod_{i=1}^{n} e(x_i P_1, y_i P_2) = \prod_{i=1}^{n} e(P_1, P_2)^{x_i y_i} = e(P_1, P_2)^{\langle\!\langle \vec{x}, \vec{y} \rangle\!\rangle}$$

Note that $\bar{e}(\mathbf{x}P_1, \mathbf{y}P_2) = 1$ iff $\langle\!\langle \mathbf{x}, \mathbf{y} \rangle\!\rangle = 0$.

Dual bases: For a DPVS \mathcal{V}_n , let $\mathbb{B} = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ and $\mathbb{B}^* = (\mathbf{b}_1^*, \dots, \mathbf{b}_n^*)$ be bases of \mathbb{V} and \mathbb{V}^* such that $\bar{e}(\mathbf{b}_i, \mathbf{b}_j^*) = 1$ for $i \neq j$ and $\bar{e}(\mathbf{b}_i, \mathbf{b}_i^*) = e(P_1, P_2)^{\psi}$ for all $i \in \{1, 2, \dots, n\}$ where ψ is an element of \mathbb{Z}_p^* . Then $(\mathbb{B}, \mathbb{B}^*)$ are called dual bases of \mathcal{V}_n . Clearly $(\mathbb{A}, \mathbb{A}^*)$ are dual bases with $\psi = 1$. Some problems in \mathbb{V} (resp. \mathbb{V}^*) (decisional subspace, for instance) are easy to solve over $(\mathbb{A}, \mathbb{A}^*)$ but computationally hard given random basis \mathbb{B} (resp. \mathbb{B}^*) unless some associated trapdoor information is provided. For instance, consider the problem of deciding whether a vector $\mathbf{v} = \sum_{i=1}^m x_i \mathbf{c}_i$ for some $1 \leq m < n$ or $\mathbf{v} \in_U \mathbb{V}$ where \mathbb{V} is given by a basis $\mathbb{C} = (\mathbf{c}_1, \dots, \mathbf{c}_n)$. If $\mathbb{C} = \mathbb{A}$, then the answer is straightforward as one can just check whether the entries of \mathbf{v} in positions m + 1 to n are zero. Instead suppose that $\mathbb{C} = \mathbb{B}$, a random basis \mathbb{B}^* of \mathbb{B} are provided, this problem becomes easy to solve - just pair a linear combination of these vectors with \mathbf{v} and check whether the result of the pairing is the identity in \mathbb{G}_T . These vectors form the trapdoor.

Generating dual bases: There is an efficient algorithm $\text{Dual}(\mathcal{V}_n)$ that returns random dual bases $(\mathbb{B}, \mathbb{B}^*)$ of \mathcal{V}_n along with a value $\psi \in \mathbb{Z}_p^{\times}$. $(\mathbb{B}, \mathbb{B}^*)$ is constructed by sampling $\mathbf{X} = (\chi_{i,j}) \xleftarrow{U} GL(n, \mathbb{Z}_p), \psi \xleftarrow{U} \mathbb{Z}_p^{\times}$ and transforming the canonical bases $(\mathbb{A}, \mathbb{A}^*)$ as $\mathbb{B} = \mathbb{A}(\psi \mathbf{X})$ and $\mathbb{B}^* = \mathbb{A}^*(\mathbf{X}^T)^{-1}$. We have $\mathbf{b}_j = \sum_{i=1}^n \chi_{i,j} \mathbf{a}_i$ and $\mathbf{b}_j^* = \sum_{i=1}^n \nu_{j,i} \mathbf{a}_i$ for $1 \le j \le n$, where $X^{-1} = (\nu_{i,j})$. It essentially applies a linear transformation to \mathbb{A} and the corresponding changes to \mathbb{A}^* to retain the duality property so that $\bar{e}(\mathbf{b}_i, \mathbf{b}_i^*) = e(P_1, P_2)^{\psi}$. Note that these vectors can be constructed given generators P_1, P_2 of $\mathbb{G}_1, \mathbb{G}_2$.

Computing scalar multiples of vectors: Suppose one wishes to compute $x\mathbf{b}_j$, for some $j \in \{1, \ldots, n\}$ and $x \in \mathbb{Z}_p$. The method of obtaining this is as follows: compute xP_1 and obtain vector $x\mathbf{a}_i$ (for all i) by just replacing P_1 with xP_1 in the *i*-th position. Now compute $x\mathbf{b}_j = \sum_{i=1}^n \chi_{i,j}(x\mathbf{a}_i)$.

3 Hardness Assumptions

Here we state subspace assumptions over asymmetric pairing-based dual pairing vector spaces. In Appendix B, we provide descriptions of discrete logarithm and decision Diffie-Hellman problems in asymmetric bilinear maps.Let $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ be an asymmetric pairing and $\mathcal{V}_n = (p, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \bar{e}, \mathbb{A}, \mathbb{A}^*)$ be a DPVS of dimension n over \mathcal{G} .

Decisional Subspace. Given a DPVS \mathcal{V}_n , define a tuple \mathcal{D} constructed according to the following distribution.

$$(\mathbb{F}, \mathbb{F}^*, \psi) \longleftarrow \mathsf{Dual}(\mathcal{V}_n), \quad \tau_1, \tau_2, \mu_1, \mu_2 \xleftarrow{\cup} \mathbb{Z}_p,$$
$$\mathbf{u}_1^* = (\mu_1 \mathbf{f}_1^* + \mu_2 \mathbf{f}_{k+1}^*), \mathbf{u}_2^* = (\mu_1 \mathbf{f}_2^* + \mu_2 \mathbf{f}_{k+2}^*), \dots, \mathbf{u}_k^* = (\mu_1 \mathbf{f}_k^* + \mu_2 \mathbf{f}_{2k}^*)$$
$$\mathbf{v}_1 = \tau_1 \mathbf{f}_1, \mathbf{v}_2 = \tau_1 \mathbf{f}_2, \dots, \mathbf{v}_k = \tau_1 \mathbf{f}_k,$$
$$\mathbf{w}_1 = (\tau_1 \mathbf{f}_1 + \tau_2 \mathbf{f}_{k+1}), \mathbf{w}_2 = (\tau_1 \mathbf{f}_2 + \tau_2 \mathbf{f}_{k+2}), \dots, \mathbf{w}_k = (\tau_1 \mathbf{f}_k + \tau_2 \mathbf{f}_{2k}),$$

 $\mathcal{D} = (\mathcal{V}_n, \mathbb{F}, \mathbf{f}_1^*, \dots, \mathbf{f}_k^*, \mathbf{f}_{2k+1}^*, \dots, \mathbf{f}_n^*, \mathbf{u}_1^*, \dots, \mathbf{u}_k^*, \mu_2),$

where k, n are positive integers with $2k \leq n$. The decisional subspace problem in \mathbb{V} (parameterised by (n,k)) is to distinguish between $(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ and $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ given \mathcal{D} .

For a PPT algorithm \mathscr{A} that outputs 0 or 1, its advantage in solving DS1(n,k) problem is defined as

$$\mathsf{Adv}_{\mathcal{V}_n}^{\mathrm{DS1}(n,k)}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{D},\mathbf{v}_1,\ldots,\mathbf{v}_k)=1] - \Pr[\mathscr{A}(\mathcal{D},\mathbf{w}_1,\ldots,\mathbf{w}_k)=1]|$$

The (ε, t) -DS1(n, k) assumption holds in \mathbb{V} if for every algorithm \mathscr{A} running in time at most t, we have $\mathsf{Adv}_{\mathcal{V}_n}^{\mathrm{DS1}(n,k)}(\mathscr{A}) \leq \varepsilon$. Similarly one can define the decisional subspace assumption in \mathbb{V}^* (DS2(n, k)). Note that the problem is easy when \mathbb{F}, \mathbb{F}^* are the canonical bases \mathbb{A}, \mathbb{A}^* .

Reductions. Let \mathcal{V}_n be a DPVS obtained from an asymmetric pairing \mathcal{G} . [CLL⁺12] provides a reduction from the DDH1 (resp. DDH2) problems in \mathcal{G} to the DS1(n,k) (resp. DS2(n,k)) problems in \mathcal{V}_n . Theorem 3.1 below summarises the result.

Theorem 3.1. If the (ε, t) -DDH1 (resp. (ε, t) -DDH2) assumption holds in \mathcal{G} , then the subspace assumption (ε', t') -DS1(n, k) (resp. (ε', t') -DS2(n, k)) holds in \mathcal{V}_n where $\varepsilon = \varepsilon'$, $t = t' + O(nk\rho)$ and ρ is the maximum time required for one scalar multiplication in either \mathbb{G}_1 or \mathbb{G}_2 .

4 HIBE from Dual Pairing Vector Spaces

The HIBE scheme we propose is built upon dual pairing vector spaces over Type-3 pairings. The construction is presented according to the definition of HIBE given in Appendix A.1 and the security is established under the ANO-IND-ID-CPA notion defined in Appendix A.2. We first provide a brief overview of the construction and proof followed by the actual construction and proof.

Overview. Let *h* denote the maximum depth of the HIBE. Identities are variable length vectors over \mathbb{Z}_p . A level- ℓ $(1 \leq \ell \leq h)$ identity lives in \mathbb{Z}_p^{ℓ} . Consider a pairing $\mathcal{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ and a dual pairing vector space $\mathcal{V}_n = (p, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \bar{e}, \mathbb{A}, \mathbb{A}^*)$ of dimension *n* over \mathcal{G} . Let $(\mathbb{B}, \mathbb{B}^*, \psi) \leftarrow \mathsf{Dual}(\mathcal{V}_n)$ and *k* be such that $2k \leq n$. Let $\mathbb{B}_1 = (\mathbf{b}_1, \dots, \mathbf{b}_k)$ and $\mathbb{B}_2 = (\mathbf{b}_{k+1}, \dots, \mathbf{b}_n)$. Similarly define \mathbb{B}_1^* and \mathbb{B}_2^* . We choose \mathbb{V} to be the ciphertext space and keys consist of vectors from \mathbb{V}^* . The reason is that elements of \mathbb{G}_1 have shorter representations compared to elements of \mathbb{G}_2 .

The dual system technique for proving security uses two types of ciphertexts and keys – normal and semifunctional. The requirement is that an attacker cannot distinguish between the two types of ciphertexts (or keys). If the attacker can generate semi-functional components on its own then it could distinguish easily. To prevent this from happening, the semi-functional space is created with some secret elements that do not reveal any information about the type of the ciphertext (or key). Also, during simulation it is essential to maintain this property even when these components are being generated from the problem instance.

In order to employ the subspace assumptions to prove security within dual system framework, the most natural method is to use \mathbb{B}_1 (resp. \mathbb{B}_1^*) to generate normal components for ciphertexts (resp. keys) and use the remaining vectors to build the semi-functional space. Suppose now that we choose to keep n and hence k, constant and use a BBG-type hash. Then the key will consist of O(h) elements, most of them needed for delegation. All these elements will have semi-functional counterparts as well. But during simulation, generating O(h) semi-functional components for a key requires O(h) amount of randomness which is unavailable from the instance. Also the constant dimension does not allow creating this randomness during simulation. The way out of this problem is to base the scheme on spaces with dimension linear in h. The maximum depth of an identity is h; hence k = h + 1 vectors are needed to hash the identity (using BBG-type hash). Since $2k \le n$ in the subspace assumptions, we need the dimension of the vector spaces to be 2h + 2.

Anonymity is obtained by keeping the \mathbb{V}^* vectors used in creating identity-hash secret. Keeping these secret, however, affects delegation. There would be no way to rerandomise keys during delegation. To overcome the problem, suitably randomised copies of vectors in the key are provided within the key. The length of the key is doubled but this enables rerandomisation during key delegation. We prove security based on subspace assumptions in \mathbb{V}, \mathbb{V}^* (DS1,DS2) which are implied by DDH assumptions in $\mathbb{G}_1, \mathbb{G}_2$. There are three stages of the reduction.

- **Reduction 1:** Here, we argue that an attacker that detects whether the challenge ciphertext is normal or semi-functional can be used to solve DS1 problem. As mentioned earlier, ciphertext is a linear combination of vectors in \mathbb{B}_1 . Using DS1 instance, each vector $\mathbf{b}_j \in \mathbb{B}_1$ is expanded into a vector in $\text{span}\langle \mathbf{b}_j, \mathbf{b}_{h+1+j} \rangle$, thus making the ciphertext semi-functional. The \mathbb{B}_2 -vectors form the semi-functional component.
- **Reduction 2:** It is is argued that changing a key from normal to semi-functional is undetectable by the attacker provided DS2 assumption holds. Here again the normal keys are expanded into semi-functional keys but in two phases. This is because the key contains two copies of the id-hash that must be created using two separate instances of DS2. The main challenge here is to show that the semi-functional components of both the key and the challenge ciphertext are properly distributed in the attacker's view.
- **Reduction 3:** In the anonymity game, the goal of an attacker is to distinguish between (semi-functional) ciphertexts corresponding to challenge 'message-identity pair' and a random message-identity pair. To show that *DPVS-AHIBE* achieves security in this sense, we provide a purely information theoretical argument once all keys and challenge ciphertexts are made semi-functional. Linear transformations are applied to the dual bases in such a way that the public parameters remain consistent and statistical distance between the distributions of information provided to the attacker before and after transformation is negligibly small.

One can see that when h = 1, the scheme is equivalent to the short IBE scheme of Chen *et.al.* [CLL⁺12]. For the reader's reference, we provide the description of this IBE in Appendix C.

4.1 Anonymous HIBE

We present the construction for our scheme *DPVS-AHIBE* = (Setup, Encrypt, KeyGen, Delegate, Decrypt).

Setup(κ): Let *h* denote the maximum depth of the HIBE. Construct a DPVS $\mathcal{V}_{2h+2} = (p, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \bar{e}, \mathbb{A}, \mathbb{A}^*)$ from a Type-3 pairing $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ (generated according to κ). Choose random dual bases $(\mathbb{B}, \mathbb{B}^*, \psi) \leftarrow \mathsf{Dual}(\mathcal{V}_{2h+2})$ and set $\mathbf{w} = \sum_{j=1}^h \mathbf{b}_{j+1}$. Pick $\alpha, (\theta_j)_{j \in [1,h]} \leftarrow^{\mathbb{U}} \mathbb{Z}_p$. Set the public parameters and master secret as follows.

 $\mathcal{PP} : (\mathcal{V}_{2h+2}, (\theta_j \mathbf{b}_1)_{j \in [1,h]}, \mathbf{w}, \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{\alpha}).$ $\mathcal{MSK}: (\alpha \mathbf{b}_2^*, \mathbf{b}_1^*, (\theta_j \mathbf{b}_{i+1}^*)_{j \in [1,h]}).$

 $\mathsf{Encrypt}(\mathcal{PP}, M, \mathbf{id} = (\mathsf{id}_1, \ldots, \mathsf{id}_\ell))$: Choose $s \xleftarrow{U} \mathbb{Z}_p$. The ciphertext consists of an element of $C_0 \in \mathbb{G}_T$ and a vector $\mathbf{c}_1 \in \mathbb{V}$ computed as follows:

$$C_0 = M \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{s\alpha}, \quad \mathbf{c}_1 = s\left(\mathbf{w} + \sum_{j=1}^{\ell} \mathsf{id}_j \theta_j \mathbf{b}_1\right)$$

 $\mathsf{KeyGen}(\mathcal{PP}, \mathcal{MSK}, \mathbf{id} = (\mathsf{id}_1, \dots, \mathsf{id}_\ell)): \text{ Pick } r \xleftarrow{U}{\subset} \mathbb{Z}_p. \text{ The secret key for } \mathbf{id}, \mathcal{SK}_{\mathbf{id}} \text{ is given by } \mathcal{SK}_{\mathbf{id}} = (\mathbf{k}_1, \mathbf{k}_2, (\mathbf{d}_{1,j}, \mathbf{d}_{2,j})_{j \in [\ell+1,h]}) \text{ where}$

$$\mathbf{k}_{1} = \alpha \mathbf{b}_{2}^{*} + r_{1} \left(\sum_{j=1}^{\ell} \operatorname{id}_{j} \theta_{j} \mathbf{b}_{j+1}^{*} - \mathbf{b}_{1}^{*} \right), \ \mathbf{d}_{1,j} = r_{1} \theta_{j} \mathbf{b}_{j+1}^{*} \text{ for } j = \ell + 1, \dots, h,$$

$$\mathbf{k}_{2} = r_{2} \left(\sum_{j=1}^{\ell} \operatorname{id}_{j} \theta_{j} \mathbf{b}_{j+1}^{*} - \mathbf{b}_{1}^{*} \right), \ \mathbf{d}_{2,j} = r_{2} \theta_{j} \mathbf{b}_{j+1}^{*} \text{ for } j = \ell + 1, \dots, h,$$

The vector \mathbf{k}_1 is used for actual decryption and the remaining vectors are used for delegation and rerandomisation.

Pick $\tilde{r} \xleftarrow{U}{\leftarrow} \mathbb{Z}_p$. The secret key for the identity $\mathbf{id} : \mathbf{id}_{\ell+1}$ is computed as follows:

 $\begin{aligned} \mathbf{k}_{1} \leftarrow \mathbf{k}_{1} + \mathsf{id}_{\ell+1}\mathbf{d}_{1,\ell+1} + \tilde{r_{1}}(\mathbf{k}_{2} + \mathsf{id}_{\ell+1}\mathbf{d}_{2,\ell+1}), \\ \mathbf{d}_{1,j} \leftarrow \mathbf{d}_{1,j} + \tilde{r_{1}}\mathbf{d}_{2,j} \text{ for } j = \ell + 2, \dots, h, \\ \mathbf{k}_{2} \leftarrow \tilde{r_{2}}(\mathbf{k}_{2} + \mathsf{id}_{\ell+1}\mathbf{d}_{2,\ell+1}), \\ \mathbf{d}_{2,j} \leftarrow \tilde{r_{2}}\mathbf{d}_{2,j} \text{ for } j = \ell + 2, \dots, h, \end{aligned}$

thus setting $r_1 \leftarrow r_1 + \tilde{r_1}r_2$, $r_2 = \tilde{r_2}r_2$ for the new secret key given by $\mathcal{SK}_{\mathbf{id}:\mathbf{id}_{\ell+1}} = (\mathbf{k}_1, \mathbf{k}_2, (\mathbf{d}_{1,j}, \mathbf{d}_{2,j})_{j \in [\ell+2,h]})$. Decrypt $(\mathcal{C}, \mathbf{id} = (\mathbf{id}_1, \dots, \mathbf{id}_\ell), \mathcal{SK}_{\mathbf{id}}, \mathcal{PP})$: Decryption is done as follows: $M = C_0/\bar{e}(\mathbf{c}_1, \mathbf{k}_1)$.

Correctness of decryption. The following computation shows the correctness of decryption.

$$\begin{split} \bar{e}(\mathbf{c}_{1},\mathbf{k}_{1}) &= \bar{e}\left(s\mathbf{w}+s\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}\mathbf{b}_{1}, \ \alpha\mathbf{b}_{2}^{*}+r\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}\mathbf{b}_{j+1}^{*}-r\mathbf{b}_{1}^{*}\right) \\ &= \bar{e}\left(s\sum_{j=1}^{h}\mathbf{b}_{j+1}+s\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}\mathbf{b}_{1}, \ \alpha\mathbf{b}_{1}^{*}+r\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}\mathbf{b}_{j+1}^{*}-r\mathbf{b}_{1}^{*}\right) \\ &= \bar{e}\left(\mathbf{b}_{2}, \ \mathbf{b}_{2}^{*}\right)^{s\alpha}\cdot\prod_{j=1}^{\ell}\bar{e}\left(s\mathbf{b}_{j+1}, r\mathbf{id}_{j}\theta_{j}\mathbf{b}_{j+1}^{*}\right)\cdot\bar{e}\left(s\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}\mathbf{b}_{1}, -r\mathbf{b}_{1}^{*}\right) \\ &= \bar{e}\left(\mathbf{b}_{2}, \ \mathbf{b}_{2}^{*}\right)^{s\alpha}\cdot\prod_{j=1}^{\ell}\bar{e}\left(\mathbf{b}_{j+1}, \ \mathbf{b}_{j+1}^{*}\right)^{rs\left(\mathsf{id}_{j}\theta_{j}\right)}\cdot\bar{e}\left(\mathbf{b}_{1}, \ \mathbf{b}_{1}^{*}\right)^{-rs\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}} \\ &= \bar{e}\left(\mathbf{b}_{2}, \ \mathbf{b}_{2}^{*}\right)^{s\alpha}\cdot\bar{e}\left(\mathbf{b}_{1}, \ \mathbf{b}_{1}^{*}\right)^{rs\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}}\cdot\bar{e}\left(\mathbf{b}_{1}, \ \mathbf{b}_{1}^{*}\right)^{-rs\sum_{j=1}^{\ell}\mathsf{id}_{j}\theta_{j}} \\ &= \bar{e}\left(\mathbf{b}_{2}, \ \mathbf{b}_{2}^{*}\right)^{s\alpha} \end{split}$$

We used two properties of dual bases here $-\bar{e}(\mathbf{b}_i, \mathbf{b}_j^*) = \mathbf{1}_T$ for $i \neq j$ and $\bar{e}(\mathbf{b}_1, \mathbf{b}_1^*) = \bar{e}(\mathbf{b}_j, \mathbf{b}_j^*)$ for all $j \in [1, 2h + 2]$.

4.2 **Proof of Security**

The security of this HIBE follows the dual system approach. This requires the definition of semi-functional ciphertexts and keys. We introduce some notation. Let $\mathbb{B} = (\mathbb{B}_1, \mathbb{B}_2)$ and $\mathbb{B}^* = (\mathbb{B}_1^*, \mathbb{B}_2^*)$ be the dual bases used in the scheme, where $\mathbb{B}_1 = (\mathbf{b}_1, \dots, \mathbf{b}_{h+1})$ and $\mathbb{B}_2 = (\mathbf{b}_{h+2}, \dots, \mathbf{b}_{2h+2})$. \mathbb{B}_1^* and \mathbb{B}_2^* are defined in a similar fashion.

Semi-functional ciphertext: Let (C_0, \mathbf{c}_1) be a normal ciphertext for some message M and identity vector id. The corresponding semi-functional ciphertext is obtained by modifying \mathbf{c}_1 as follows:

$$\vec{x} \xleftarrow{U} \mathbb{Z}_p^{h+1}, \quad \mathbf{c}_1 \leftarrow \mathbf{c}_1 + \mathbb{B}_2 \vec{x}$$

Semi-functional key: For an identity vector **id** of length ℓ , let $(\mathbf{k}_1, \mathbf{k}_2, (\mathbf{d}_{1,j}, \mathbf{d}_{2,j})_{j \in [\ell+1,h]})$ be a normal key. A semi-functional key for **id** is computed as follows:

$$\vec{y}_1, \vec{y}_2, (\vec{z}_{1,j}, \vec{z}_{2,j})_{j \in [\ell+1,h]} \xleftarrow{\mathrm{U}} \mathbb{Z}_p^{h+1},$$

$$\begin{aligned} \mathbf{k}_1 \leftarrow \mathbf{k}_1 + \mathbb{B}_2^* \vec{y}_1, \quad \mathbf{d}_{1,j} \leftarrow \mathbf{d}_{1,j} + \mathbb{B}_2^* \vec{z}_{1,j} & \text{for } j \in [\ell+1,h], \\ \mathbf{k}_2 \leftarrow \mathbf{k}_2 + \mathbb{B}_2^* \vec{y}_2, \quad \mathbf{d}_{2,j} \leftarrow \mathbf{d}_{2,j} + \mathbb{B}_2^* \vec{z}_{2,j} & \text{for } j \in [\ell+1,h]. \end{aligned}$$

The set of vectors in \mathbb{B}_2 and \mathbb{B}_2^* form the semi-functional space. The scheme itself does not use them.

Partial Semi-functional Key: Similar to semi-functional keys except that $\vec{y}_2 = \vec{z}_{2,\ell+1} = \cdots = \vec{z}_{2,h} = \vec{0}$. In other words, only \mathbf{k}_1 , $(\mathbf{d}_{1,j})$ have semi-functional components.

Nominal Semi-functionality: Decryption of a semi-functional ciphertext C with a semi-functional key $S\mathcal{K}_{id}$ for an identity id will succeed only when $\langle\!\langle \vec{x}, \vec{y}_1 \rangle\!\rangle \equiv 0 \pmod{p}$. Such a pair $(C, S\mathcal{K}_{id})$ is called *nominally semi-functional*.

We now present two lemmas that will be useful in the reductions that follow. These are standard results and hence we omit the proofs.

Lemma 4.1. Let $S = \{(\vec{u}, \vec{v}) : \vec{u}, \vec{v} \in \mathbb{Z}_p^n \text{ and } \langle \langle \vec{u}, \vec{v} \rangle \rangle \neq 0 \}$. If $\mathbf{X} \xleftarrow{U}{=} \mathbb{Z}_p^{n \times n}$ is invertible, then for all $(\vec{x}, \vec{v}) \in S$, for all $(\vec{r}, \vec{w}) \in S$, and for $\mu, \tau \xleftarrow{U}{=} \mathbb{Z}_p^{\times}$,

$$\Pr\left[\left(\mu \mathbf{X}^{-1} \vec{u} = \vec{r}\right) \land \left(\tau \mathbf{X}^{T} \vec{v} = \vec{w}\right)\right] = \frac{1}{|\mathcal{S}|}$$

where $|\mathcal{S}| = (p^n - 1)(p^n - p^{n-1})$ and the probability is over the choice of \mathbf{X} , μ and τ .

Essentially, the lemma suggests that $\mu \mathbf{U}^T \vec{u}$ and $\tau \mathbf{X}^T \vec{v}$ are uniformly and independently distributed unless $\langle \langle \vec{r}, \vec{w} \rangle = 0$. This condition holds only when $\langle \langle \vec{x}, \vec{v} \rangle = 0$ which happens with probability 1/p.

Lemma 4.2. Let $\mathbf{X} \stackrel{\mathrm{U}}{\longleftarrow} \mathbb{Z}_p^{n \times n}$. Then

$$\Pr[\mathbf{X} \in GL(n, \mathbb{Z}_p)] = \frac{\prod_{i=1}^n (p^n - p^{i-1})}{p^{n^2}}.$$

We now present the security proof for $\mathcal{DPVS}-\mathcal{AHIBE}$. The shorthand $\mathcal{H}_{\ell}(\mathbf{id})$ will be used to denote the quantity $\sum_{j=1}^{\ell} \mathrm{id}_{j} \theta_{j}$ where $\mathbf{id} = (\mathrm{id}_{1}, \ldots, \mathrm{id}_{\ell})$. Let $\gamma_{n}(p) = \prod_{i=1}^{n} (p^{n} - p^{i-1})/p^{n^{2}}$.

Theorem 4.1. If the $(\varepsilon_{\text{DDH1}}, t_{\text{DDH1}})$ -DDH1, $(\varepsilon_{\text{DDH2}}, t_{\text{DDH2}})$ -DDH2 assumptions hold and ADLP is $(\varepsilon_{\text{ADLP}}, t_{\text{ADLP}})$ -hard in \mathcal{G} , then DPVS-AHIBE is (ε, t, q) -ANO-IND-ID-CPA secure where

$$\varepsilon \le 2q\varepsilon_{\text{ADLP}} + \frac{\varepsilon_{\text{DDH1}}}{\gamma_{h+1}(p)} + \frac{2q\varepsilon_{\text{DDH2}}}{\gamma_{h+1}(p)} + \frac{1}{p}$$

and $t_{\text{DDH1}} = t + O(h^2 \rho)$, $t_{\text{DDH2}} = t + O(qh^2 \rho)$, $t_{\text{ADLP}} = t + O(h^2 \rho)$ where ρ is the maximum time for one scalar multiplication in either of $\mathbb{G}_1, \mathbb{G}_2$.

Proof. We provide a hybrid argument over a sequence of 2q + 4 games, where q is the number of key extract queries made by the adversary. Let G_{actual} denote the actual IBE CPA-security game ano-ind-cpa described in Appendix A.2. G_{actual} is same as $G_{restricted}$ except that the following restriction is added to the game: $\mathcal{H}_{\hat{\ell}}(\hat{\mathbf{id}}_{\beta}) \notin \mathcal{H}_{\ell_k}(\mathbf{id}_k) \pmod{p}$ for all $k \in \{1, \ldots, q\}$ where $\hat{\mathbf{id}}_{\beta} (\beta = 0, 1)$ are the challenge identities and \mathbf{id}_k is the identity provided by the adversary in the k-th key extract query. Here $\hat{\ell}_{\beta}$ and ℓ_k are the lengths of the identity tuples $\hat{\mathbf{id}}_{\beta}$ and \mathbf{id}_k . $G_{0,1}$ is just like $G_{restricted}$ except that the challenge ciphertext is a semi-functional encryption of the chosen message to the corresponding identity vector. Define $G_{k,0}$ for $k \in [1,q]$ similar to $G_{0,1}$ except that the first k-1 keys returned to the adversary are semi-functional, *k*-th key is partial semi-functional and the rest are normal. $G_{k,1}$ proceeds the same way as $G_{k,0}$ with one difference – *k*-th key is fully semi-functional. Let G_{final} be defined similar to $G_{q,1}$ except that now the challenge ciphertext is a semi-functional encryption of a random message to a random identity vector. Let X_{\Box} denote the event that the adversary wins in G_{\Box} . Note that, in G_{final} , the challenge ciphertext is an encryption of a random message and hence bit β is statistically hidden from the adversary's view implying that $\Pr[X_{final}] = 1/2$.

Using lemmas 4.3, 4.4, 4.5,4.6 and 4.7, we have for any *t*-time adversary \mathscr{A} against DPVS-AHIBE in the ano-ind-cpa game, there is an algorithm \mathscr{B} such that

$$\begin{aligned} \mathsf{Adv}_{\mathcal{DPVS-AHIBE}}^{\mathsf{ano-ind-cpa}}(\mathscr{A}) &= \left| \Pr[X_{actual}] - \frac{1}{2} \right| \\ &= \left| \Pr[X_{actual}] - \Pr[X_{final}] \right| \\ &\leq \left| \Pr[X_{actual}] - \Pr[X_{restricted}] \right| + \left| \Pr[X_{restricted}] - \Pr[X_{0,1}] \right| \\ &+ \sum_{k=1}^{q} \left(\left| \Pr[X_{k-1,1}] - \Pr[X_{k,0}] \right| \right) + \sum_{k=1}^{q} \left(\left| \Pr[X_{k,0}] - \Pr[X_{k,1}] \right| \right) + \left| \Pr[X_{q}] - \Pr[X_{final}] \right| \\ &\leq 2q\varepsilon_{\mathrm{ADLP}} + \frac{\varepsilon_{\mathrm{DS1}(n,k)}}{\gamma_{h+1}(p)} + 2q\left(\frac{\varepsilon_{\mathrm{DS2}(n,k)}}{\gamma_{h+1}(p)}\right) + \frac{1}{p} \end{aligned}$$

where n = 2h + 2 and k = h + 1. The theorem now follows directly from Theorems 3.1 since n and k are linear in h in the DS1(n, k), DS2(n, k) problems we consider.

In all the following lemmas, \mathscr{B} is the simulator and \mathscr{A} is an **ano-ind-cpa** attacker against DPVS-AHIBE making q key extract queries. Also, we consider subspace assumptions with n = 2h + 2 and k = h + 1.

Lemma 4.3. $|\Pr[X_{actual}] - \Pr[X_{restricted}]| \le 2q\varepsilon_{ADLP}$.

We refer to [BGG94] for main ideas underlying the proof and only provide details specific to our construction. Appendix D contains the details.

Lemma 4.4. $|\Pr[X_{restricted}] - \Pr[X_{0,1}]| \le \varepsilon_{DS1(2h+2,h+1)} / \gamma_{h+1}(p).$

Proof. The algorithm \mathscr{B} receives $(\mathcal{V}_{2h+2}, \mathbb{F}, \mathbb{F}_1^*, \mathbf{u}_1^*, \dots, \mathbf{u}_{h+1}^*, \mu_2, \mathbf{t}_1, \dots, \mathbf{t}_{h+1})$ as an instance of DS1(2h+2, h+1) problem. Here $\mathbf{u}_j = (\mu_1 \mathbf{f}_j^* + \mu_2 \mathbf{f}_{h+1+j}^*)$ for $j = 1, \dots, h+1$. If $\mathbf{t}_j = \tau_1 \mathbf{f}_j$ for all $j \in [1, h+1]$, we call them "real" and when they are distributed as $\tau_1 \mathbf{f}_j + \tau_2 \mathbf{f}_{h+1+j}$, we call them "random". So \mathscr{B} 's task is to decide whether $(\mathbf{t}_j)_{j \in [1,h+1]}$ are real or random. We describe how \mathscr{B} simulates each phase of the security game.

Setup: \mathscr{B} first chooses $\mathbf{X} \xleftarrow{U} \mathbb{Z}_p^{(h+1)\times(h+1)}$ and if $\mathbf{X} \notin GL(h+1,\mathbb{Z}_p)$, aborts the game and returns a random bit. Otherwise, it sets the dual bases $(\mathbb{B}, \mathbb{B}^*)$ of the HIBE scheme as follows.

$$\mathbb{B}_1 = \mathbb{F}_1, \quad \mathbb{B}_2 = \mathbb{F}_2 \mathbf{X}, \quad \mathbb{B}_1^* = \mathbb{F}_1^*, \quad \mathbb{B}_2^* = \mathbb{F}_2^* (\mathbf{X}^{-1})^T.$$

It then chooses $\alpha, \theta_1, \ldots, \theta_h \xleftarrow{U} \mathbb{Z}_p$ and provides the following public parameters to \mathscr{A} .

$$\mathcal{PP} = (\mathcal{V}_{2h+2}, (\theta_j \mathbf{b}_1)_{j \in [1,h]}, \mathbf{w}, \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{\alpha})$$

 \mathscr{B} knows the master secret key $\mathcal{MSK} = (\alpha \mathbf{b}_2^*, \mathbf{b}_1^*, (\theta_j \mathbf{b}_{j+1}^*)_{j \in [1,h]})$ using which normal keys for any identity vector can be created.

Phase 1: \mathscr{A} makes a number of key extract queries. \mathscr{B} returns a normal key generated using the KeyGen algorithm for every such query.

Challenge: \mathscr{B} receives the two pairs $(M_0, \widehat{\mathbf{id}}_0)$ and $(M_1, \widehat{\mathbf{id}}_1)$ from \mathscr{A} . It chooses $\beta \xleftarrow{U} \{0, 1\}$ and encrypts M_β to $\widehat{\mathbf{id}}_\beta = (\widehat{\mathbf{id}}_1, \dots, \widehat{\mathbf{id}}_{\widehat{\ell}})$ as:

$$C_0 = M_\beta \cdot \bar{e}(\mathbf{t}_2, \mathbf{b}_2^*)^\alpha, \qquad \mathbf{c}_1 = \sum_{j=1}^h \mathbf{t}_{j+1} + \left(\sum_{j=1}^{\widehat{\ell}} \widehat{\mathsf{id}}_j \theta_j\right) \mathbf{t}_1,$$

where $s = \tau_1$ is implicitly set. Ciphertext $\widehat{\mathcal{C}} = (C_0, \mathbf{c}_1)$ is returned to \mathscr{A} .

Now if \mathbf{t}_j 's are real, $\widehat{\mathcal{C}}$ is distributed as a normal ciphertext and in this case \mathscr{B} simulates $\mathsf{G}_{restricted}$. The computation shown below illustrates this fact.

$$\mathbf{c}_{1} = \sum_{j=1}^{h} \mathbf{t}_{j+1} + \left(\sum_{j=1}^{\widehat{\ell}} \widehat{\mathbf{id}}_{j} \theta_{j}\right) \mathbf{t}_{1} = \sum_{j=1}^{h} \tau_{1} \mathbf{f}_{j+1} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}) \tau_{1} \mathbf{f}_{1} = \tau_{1} \left(\sum_{j=1}^{h} \mathbf{b}_{j+1} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}) \mathbf{b}_{1}\right) = s \left(\mathbf{w} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}) \mathbf{b}_{1}\right).$$

Next consider the case when $\mathbf{t}_j = \tau_1 \mathbf{f}_j + \tau_2 \mathbf{f}_{h+1+j}$. We have

$$\mathbf{c}_{1} = \sum_{j=1}^{h} \mathbf{t}_{j+1} + \left(\sum_{j=1}^{\widehat{\ell}} \widehat{\mathbf{id}}_{j} \theta_{j}\right) \mathbf{t}_{1}$$

$$= \sum_{j=1}^{h} \tau_{1} \mathbf{f}_{j+1} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}) \tau_{1} \mathbf{f}_{1} + \sum_{j=1}^{h} \tau_{2} \mathbf{f}_{h+j+2} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}) \tau_{2} \mathbf{f}_{h+2}$$

$$= \tau_{1} \left(\sum_{j=1}^{h} \mathbf{b}_{j+1} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}) \mathbf{b}_{1}\right) + \tau_{2} \mathbb{F}_{2} (\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}), \underbrace{1, \dots, 1}_{h})^{T}$$

$$= s \left(\mathbf{w} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}) \mathbf{b}_{1}\right) + \tau_{2} \mathbb{B}_{2} \mathbf{X}^{-1} (\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}), \underbrace{1, \dots, 1}_{h})^{T}$$

thus setting $\vec{x} = \tau_2 \mathbf{X}^{-1}(\mathcal{H}_{\hat{\ell}}(\widehat{\mathbf{id}}_{\beta}), 1, \dots, 1)^T$ which has the appropriate distribution due to the choice of \mathbf{X} and the fact that all information provided to \mathscr{A} are independent of \mathbf{X} . \mathscr{B} 's simulation is, therefore, perfect.

Phase 2: As in first phase, \mathscr{B} returns a normal key for every query.

Guess: \mathscr{A} returns its guess β' .

If the adversary wins the game then \mathscr{B} returns 1; otherwise it returns 0. Let Y_{real} and Y_{random} denote the events that \mathscr{B} returns 1 when \mathbf{t}_j 's are real and random respectively. Let the event abort denote the event that \mathscr{B} aborts the game. Then, we have

$$\Pr[Y_{real}] = \Pr[Y_{real} | \text{abort}] \Pr[\text{abort}] + \Pr[Y_{real} | \overline{\text{abort}}] \Pr[\overline{\text{abort}}]$$

$$= \frac{1}{2} \Pr[\text{abort}] + \Pr[X_{restricted}] \Pr[\overline{\text{abort}}], \qquad (1)$$

$$\Pr[Y_{random}] = \Pr[Y_{random} | \text{abort}] \Pr[\overline{\text{abort}}] + \Pr[Y_{random} | \overline{\text{abort}}] \Pr[\overline{\text{abort}}]$$

$$= \frac{1}{2} \Pr[\text{abort}] + \Pr[X_{0,1}] \Pr[\overline{\text{abort}}] \qquad (2)$$

Recall that \mathscr{B} aborts when **X** is not invertible. From Lemma 4.2 it follows that $\Pr[\overline{\mathsf{abort}}] = \gamma_{h+1}(p)$. Subtracting (2) from (1) and taking absolute values, we obtain

$$|\Pr[Y_{real}] - \Pr[Y_{random}]| = |\Pr[X_{restricted}] - \Pr[X_{0,1}]| \cdot \Pr[\overline{\mathsf{abort}}]$$

$$\geq |\Pr[X_{restricted}] - \Pr[X_{0,1}]|\gamma_{h+1}(p).$$

 $\mathcal B$'s advantage in solving the DS1 problem is given by

$$\operatorname{\mathsf{Adv}}_{\mathcal{V}_{2h+2}}^{\operatorname{DS1}(2h+2,h+1)}(\mathscr{B}) = |\operatorname{Pr}[Y_{real}] - \operatorname{Pr}[Y_{random}]| \ge |\operatorname{Pr}[X_{restricted}] - \operatorname{Pr}[X_{0,1}]|\gamma_{h+1}(p).$$

Lemma 4.5. $|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \le \varepsilon_{\text{DS2}(2h+2,h+1)} / \gamma_{h+1}(p).$

Proof. \mathscr{B} gets an instance $(\mathcal{V}_{2h+2}, \mathbb{F}_1, \mathbb{F}^*, \mathbf{u}_1, \dots, \mathbf{u}_{h+1}, \mu_2, \mathbf{t}_1^*, \dots, \mathbf{t}_{h+1}^*)$ of DS2(2h + 2, h + 1), where $\mathbf{u}_j = (\mu_1 \mathbf{f}_j + \mu_2 \mathbf{f}_{h+1+j})$. \mathscr{B} has to determine whether (\mathbf{t}_j^*) are real or random – i.e., whether they are distributed as $(\tau_1 \mathbf{f}_j^*)$ or $(\tau_1 \mathbf{f}_j^* + \tau_2 \mathbf{f}_{h+1+j}^*)$.

Setup: \mathscr{B} samples $\mathbf{X} = (\chi_{i,j}) \xleftarrow{U}{\mathbb{Z}_p^{(h+1)\times(h+1)}}$. If \mathbf{X} is not invertible, it aborts and returns a random bit; otherwise it proceeds to set the dual bases $(\mathbb{B}, \mathbb{B}^*)$ of the HIBE scheme as follows.

$$\mathbb{B}_1 = \mathbb{F}_1, \quad \mathbb{B}_2 = \mathbb{F}_2 \mathbf{X}, \quad \mathbb{B}_1^* = \mathbb{F}_1^*, \quad \mathbb{B}_2^* = \mathbb{F}_2^* (\mathbf{X}^{-1})^T.$$

 \mathscr{B} then picks $\alpha, \theta_1, \ldots, \theta_h \xleftarrow{U} \mathbb{Z}_p$ and sends the public parameters \mathcal{PP} to \mathscr{A} . \mathcal{MSK} is known to \mathscr{B} and hence normal keys can be created. \mathscr{B} knows \mathbb{F}_2^* and hence \mathbb{B}_2^* using which it can generate semi-functional keys.

Phases 1 and 2: Let $\mathbf{id}_1, \mathbf{id}_2, \ldots, \mathbf{id}_q$ denote the identities for which \mathscr{A} requests the secret keys. For i > k, \mathscr{B} generates a normal key for \mathbf{id}_i using the master secret and returns the resulting key to \mathscr{A} . For i < k, \mathscr{B} first generates a normal key and then modifies it appropriately to obtain a semi-functional key. The resulting key is returned to \mathscr{A} . In case of the k-th query (i.e., i = k), \mathscr{B} generates the key as follows. Let $\mathbf{id}_k = (\mathbf{id}_1, \ldots, \mathbf{id}_\ell)$.

$$\mathbf{k}_{1} = \alpha \mathbf{f}_{2}^{*} + \left(\sum_{j=1}^{\ell} \mathsf{id}_{j} \theta_{j} \mathbf{t}_{j+1}^{*} - \mathbf{t}_{1}^{*}\right), \quad \mathbf{d}_{1,\ell+1} = \theta_{\ell+1} \mathbf{t}_{\ell+2}^{*}, \ \dots, \ \mathbf{d}_{1,h} = \theta_{h} \mathbf{t}_{h+1}^{*}.$$

 \mathbf{k}_{2} , $(\mathbf{d}_{2,j})$ are created normally, as in the KeyGen algorithm. The key $\mathcal{SK}_{\mathbf{id}_{k}} = (\mathbf{k}_{1}, \mathbf{k}_{2}, (\mathbf{d}_{1,j}, \mathbf{d}_{2,j})_{j \in [\ell+1,h]})$ is returned to \mathscr{A} .

If \mathbf{t}_{j}^{*} are real, then it is clear that $\mathcal{SK}_{\mathbf{id}_{k}}$ is normal with $r_{1} = \tau_{1}$ being set implicitly. In this case \mathscr{B} simulates $\mathbf{G}_{k-1,1}$. Otherwise \mathbf{t}_{j}^{*} are random and the key will be semi-functional as justified below.

$$\begin{aligned} \mathbf{k}_{1} &= \alpha \mathbf{f}_{2}^{*} + \left(\sum_{j=1}^{\ell} \mathrm{id}_{j} \theta_{j} \mathbf{t}_{j+1}^{*} - \mathbf{t}_{1}^{*}\right) \\ &= \alpha \mathbf{b}_{2}^{*} + \left(\sum_{j=1}^{\ell} \mathrm{id}_{j} \theta_{j} \tau_{1} \mathbf{f}_{j+1}^{*} - \tau_{1} \mathbf{f}_{1}^{*}\right) + \left(\sum_{j=1}^{\ell} \mathrm{id}_{j} \theta_{j} \tau_{2} \mathbf{f}_{h+j+2}^{*} - \tau_{2} \mathbf{f}_{h+2}^{*}\right) \\ &= \alpha \mathbf{b}_{2}^{*} + r_{1} \left(\sum_{j=1}^{\ell} \mathrm{id}_{j} \theta_{j} \mathbf{b}_{j+1}^{*} - \mathbf{b}_{1}^{*}\right) + \tau_{2} \mathbb{F}_{2}^{*} (-1, \theta_{1} \mathrm{id}_{1}, \dots, \theta_{\ell} \mathrm{id}_{\ell}, \underbrace{0, \dots, 0}_{h-\ell-1})^{T} \\ &= \alpha \mathbf{b}_{2}^{*} + r \left(\sum_{j=1}^{\ell} \mathrm{id}_{j} \theta_{j} \mathbf{b}_{j+1}^{*} - \mathbf{b}_{1}^{*}\right) + \tau_{2} \mathbb{B}_{2}^{*} \mathbf{X}^{T} (-1, \theta_{1} \mathrm{id}_{1}, \dots, \theta_{\ell} \mathrm{id}_{\ell}, \underbrace{0, \dots, 0}_{h-\ell-1})^{T} \\ &\mathbf{d}_{1,j} = \theta_{j} \mathbf{t}_{j+1}^{*} = \theta_{j} (\tau_{1} \mathbf{f}_{j+1}^{*} + \tau_{2} \mathbf{f}_{h+j+2}^{*}) = r_{1} \theta_{j} \mathbf{b}_{j+1}^{*} + \tau_{2} \theta_{j} \sum_{i=1}^{h+1} \chi_{j+1,i} \mathbf{b}_{h+1+i}^{*}, \end{aligned}$$

setting $r_1 = \tau_1$, $\vec{y}_1 = \tau_2 \mathbf{X}^T (-1, \theta_1 \operatorname{id}_1, \dots, \theta_\ell \operatorname{id}_\ell, 0, \dots, 0)^T$, $\vec{z}_{1,j} = \theta_j \tau_2 (\chi_{j+1,1}, \dots, \chi_{j+1,h+1})$ for $\ell + 1 \leq j \leq h$. Consequently, \mathscr{B} simulates $\mathsf{G}_{k,0}$.

Challenge: \mathscr{B} receives $(M_0, \widehat{\mathbf{id}}_0), (M_1, \widehat{\mathbf{id}}_1)$ from \mathscr{A} . It chooses $\beta \in \{0, 1\}$ at random. Let $\widehat{\mathbf{id}}_{\beta} = (\widehat{\mathbf{id}}_1, \ldots, \widehat{\mathbf{id}}_{\widehat{\ell}})$. \mathscr{B} encrypts M_{β} to $\widehat{\mathbf{id}}$ as

$$C_0 = M_\beta \cdot \bar{e}(\mathbf{u}_2, \mathbf{b}_2^*)^\alpha, \ \mathbf{c}_1 = \sum_{j=1}^h \mathbf{u}_{j+1} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_\beta)\mathbf{u}_1$$

and returns $\widehat{\mathcal{C}} = (C_0, \mathbf{c}_1)$ to \mathscr{A} . Component \mathbf{c}_1 is properly distributed since

$$\mathbf{c}_{1} = \sum_{j=1}^{h} \mathbf{u}_{j+1} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta})\mathbf{u}_{1}$$

$$= \mu_{1} \sum_{j=1}^{h} \mathbf{f}_{j+1} + \mu_{1}\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta})\mathbf{f}_{1} + \mu_{2} \sum_{j=1}^{h} \mathbf{f}_{h+j+2} + \mu_{2}\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta})\mathbf{f}_{h+2}$$

$$= \mu_{1} \sum_{j=1}^{h} \mathbf{b}_{j+1} + \mu_{1}\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta})\mathbf{b}_{1} + \mu_{2}\mathbb{F}_{2}(\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}), \underbrace{1, \dots, 1}_{h})$$

$$= s\mathbf{w} + s\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta})\mathbf{b}_{1} + \mu_{2}\mathbb{B}_{2}\mathbf{X}^{-1}(\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}), \underbrace{1, \dots, 1}_{h})^{T}$$

implicitly setting $s = \mu_1$ and $\vec{x} = \mu_2 \mathbf{X}^{-1} (\mathcal{H}_{\hat{\ell}}(\widehat{\mathbf{id}}_\beta), 1, \dots, 1)^T$.

Guess: \mathscr{A} returns its guess β' of β .

We argue that all information provided to \mathscr{A} are properly distributed. This is indeed the case when (\mathbf{t}_{j}^{*}) are real i.e., in G_{k-1} . Let us take a look at the joint distribution of all the scalars in \mathscr{A} 's view when G_{k} is being simulated.

Due to the choice of the θ_j 's, **X** and the distribution of τ_2 , $\vec{z}_{1,j}$'s will be uniformly distributed in \mathbb{Z}_p^{h+1} . The (j+1)-st row of **X**, $\vec{\chi}_{j+1}$, provides the randomness required to generate $\vec{z}_{1,j}$. Also, $\vec{z}_{1,j}$'s are independent of the first row of **X**, which provides the randomness for \vec{x} and \vec{y}_1 . Therefore the semi-functional components of $\mathbf{d}_{1,j}$'s will be properly distributed. Next we show that in \mathbf{G}_k , the semi-functional components of $\mathcal{SK}_{\mathbf{id}_k}$ and $\widehat{\mathcal{C}}$ are properly distributed in \mathscr{A} 's view. For this, we need only argue that the co-efficient vectors \vec{x} and \vec{y}_1 given by

$$\vec{x} = \mu_2 \mathbf{X}^{-1} (\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_\beta), 1, \dots, 1)^T \text{ and } \vec{y}_1 = \tau_2 \mathbf{X}^T (-1, \theta_1 \mathrm{id}_1, \dots, \theta_\ell \mathrm{id}_\ell, 0, \dots, 0)^T$$

are uniformly and independently distributed. Note that all the information provided to the adversary except for the challenge ciphertext and the response to the k-th key extraction query are independent of the matrix **X**. Also, observe that $\mathbf{id}_k \neq \mathbf{id}_\beta$ and the quantities $\theta_1, \ldots, \theta_h$ are uniformly distributed over \mathbb{Z}_p . Hence, by Lemma 4.1 it follows that the coefficient vectors are uniformly and independently distributed unless $\langle\!\langle \vec{x}, \vec{y}_1 \rangle\!\rangle = \vec{x}^T \vec{y}_1 = 0 \pmod{p}$ i.e., $\mu_2 \tau_2(\mathcal{H}_{\hat{\ell}}(\mathbf{id}_\beta) - \sum_{j=1}^h \theta_j \mathbf{id}_j) = 0 \pmod{p}$ i.e., $\mathcal{H}_{\ell}(\mathbf{id}_k) \equiv \mathcal{H}_{\hat{\ell}}(\mathbf{id}_\beta)$ (mod p). We have ruled out this possibility in Lemma 4.3.

Observe that, if \mathscr{B} tries to generate a semi-functional ciphertext for \mathbf{id}_k (in order to determine whether $\mathcal{SK}_{\mathsf{id}_k}$ is semi-functional), the coefficient vectors of $\mathbb{B}_2, \mathbb{B}_2^*$ for the ciphertext-key pair will end up being orthogonal thus resulting in a nominal semi-functionality. This provides no information to \mathscr{B} of whether $\mathcal{SK}_{\mathsf{id}_k}$ is semi-functional and hence about the distribution of \mathbf{t}_j^* .

It remains to bound $\operatorname{Adv}_{\mathcal{V}_{2h+2}}^{\operatorname{DS2}(2h+2,h+1)}(\mathscr{B})$. If $\beta = \beta'$, then \mathscr{B} returns 1; otherwise it returns 0. Let Y_{real} and Y_{random} denote the events that \mathscr{B} returns 1 when (\mathbf{t}_j^*) are real and random respectively. \mathscr{B} 's

outputs a bit (depending on \mathscr{A} 's guess) as long as it does not abort. It is straightforward to see that $|\Pr[Y_{real}] - \Pr[Y_{random}]| = |\Pr[X_{k-1,1}] - \Pr[X_{k,0}]| \cdot \Pr[\overline{\mathsf{abort}}]$. We know from Lemma 4.2 that $\Pr[\overline{\mathsf{abort}}] = \gamma_{h+1}(p)$. Therefore, we have

$$\mathsf{Adv}_{\mathcal{V}_{2h+2}}^{\mathrm{DS2}(2h+2,h+1)}(\mathscr{B}) = |\Pr[Y_{real}] - \Pr[Y_{random}]| \ge \gamma_{h+1}(p) \left|\Pr[X_{k-1,1}] - \Pr[X_{k,0}]\right|.$$

from which the lemma follows.

Lemma 4.6. $|\Pr[X_{k,0}] - \Pr[X_{k,1}]| \le \varepsilon_{DS2(2h+2,h+1)} / \gamma_{h+1}(p).$

The proof is similar to that of the previous lemma. The only difference is that $(\mathbf{k}_1, (\mathbf{d}_{1,j}))$ are made semi-functional and the instance is embedded in $(\mathbf{k}_2, (\mathbf{d}_{2,j}))$.

Lemma 4.7. $|\Pr[X_{q,1}] - \Pr[X_{final}]| \le \frac{1}{p}$.

Proof. Let $\mathcal{D}_{q,1}$ and \mathcal{D}_{final} denote the distribution of all information provided to \mathscr{A} in $\mathsf{G}_{q,1}$ and G_{final} respectively.

$$\mathcal{D}_{q,1} = (\mathcal{PP}, \widehat{\mathcal{C}}, (\mathcal{SK}_{\mathbf{id}_k})_{k=1,\ldots,q})$$

where $\widehat{\mathcal{C}}$ is a semi-functional encryption of M_{β} under \widehat{id}_{β} for some $\beta \in_{\mathrm{U}} \{0,1\}$ and \mathcal{SK}_{id_k} is a semi-functional key for id_k .

$$\mathcal{D}_{final} = (\mathcal{PP}, \mathcal{C}', (\mathcal{SK}_{\mathbf{id}_k})_{k=1,...,q})$$

where \mathcal{C}' is a semi-functional encryption of a random message to a random identity vector. The goal here is to show that $\mathcal{D}_{q,1}$ and \mathcal{D}_{final} are statistically indistinguishable except with probability 1/p. A change of basis transformation is applied to the dual bases $(\mathbb{B}, \mathbb{B}^*)$ in $G_{q,1}$ keeping the public parameters consistent. Then it is argued that \mathscr{A} 's view under the new bases is distinguishable from its view before the transformation with probability at most 1/p. Details follow.

Let $\mathbb{B} = (\mathbb{B}_1, \mathbb{B}_2)$ and $\mathbb{B}^* = (\mathbb{B}_1^*, \mathbb{B}_2^*)$. Define new dual bases $\mathbb{F} = (\mathbb{F}_1, \mathbb{F}_2)$ and $\mathbb{F}^* = (\mathbb{F}_1^*, \mathbb{F}_2^*)$ as:

$$\vec{\chi}_1, \vec{\chi}_2 \xleftarrow{U} \mathbb{Z}_p^{h+1}$$

$$\mathbf{Y} = (\vec{\chi}_1, \underbrace{\vec{\chi}_2, \vec{\chi}_2, \dots, \vec{\chi}_2}_h) \in \mathbb{Z}_p^{(h+1) \times (h+1)}$$

$$\mathbb{F}_1 = \mathbb{B}_1, \quad \mathbb{F}_2 = \mathbb{B}_2 + \mathbb{B}_1 \mathbf{Y}, \quad \mathbb{F}_1^* = \mathbb{B}_1^* - \mathbb{B}_2^* \mathbf{Y}, \quad \mathbb{F}_2^* = \mathbb{B}_2^*.$$

Clearly $(\mathbb{F}, \mathbb{F}^*)$ are dual bases and properly distributed.

Information provided to the adversary (i.e., public parameters, challenge ciphertext and responses to key extract queries) in $G_{q,1}$ when expressed over bases $(\mathbb{B}, \mathbb{B}^*)$ will be

$$\mathcal{PP} = (\mathcal{G}, \mathbf{w}, \theta_1 \mathbf{b}_1, \dots, \theta_h \mathbf{b}_1, \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{\alpha}),$$
$$\widehat{\mathcal{C}} = \left(C_0 = M_\beta \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{s\alpha}, \mathbf{c}_1 = s(\mathbf{w} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_\beta)\mathbf{b}_1) + \mathbb{B}_2 \vec{x}\right),$$
$$\mathcal{SK}_{\mathbf{id}_k} = \left(\mathbf{k}_1^{(k)}, \mathbf{d}_{1,\ell_k+1}^{(k)}, \dots, \mathbf{d}_{1,h}^{(k)}, \mathbf{k}_2^{(k)}, \mathbf{d}_{2,\ell_k+1}^{(k)}, \dots, \mathbf{d}_{2,h}^{(k)}\right) \text{ for } k = 1, \dots, q_k$$

with

$$\mathbf{k}_{1}^{(k)} = \alpha \mathbf{b}_{2}^{*} + r_{1}^{(k)} (\vec{\gamma}^{(k)})_{\mathbb{B}_{1}^{*}} + (\vec{y}_{1}^{(k)})_{\mathbb{B}_{2}^{*}},$$

$$\mathbf{d}_{1,j}^{(k)} = r_{1}^{(k)} \theta_{j} \mathbf{b}_{j+1}^{*} + (\vec{z}_{j,1}^{(k)})_{\mathbb{B}_{2}^{*}} \text{ for } \ell_{k} + 1 \le j \le h,$$

$$\begin{aligned} \mathbf{k}_{2}^{(k)} &= r_{2}^{(k)} (\vec{\gamma}^{(k)})_{\mathbb{B}_{1}^{*}} + (\vec{y}_{2}^{(k)})_{\mathbb{B}_{2}^{*}}, \\ \mathbf{d}_{2,j}^{(k)} &= r_{2}^{(k)} \theta_{j} \mathbf{b}_{j+1}^{*} + (\vec{z}_{j,2}^{(k)})_{\mathbb{B}_{2}^{*}} \text{ for } \ell_{k} + 1 \leq j \leq h \end{aligned}$$

where $\vec{\gamma}^{(k)} = (-1, \theta_1 \mathsf{id}_{k,1}, \dots, \theta_{\ell_k} \mathsf{id}_{k,\ell_k}, \underbrace{0, \dots, 0}_{h-\ell_k})$. The same information, when expressed over bases $(\mathbb{F}, \mathbb{F}^*)$,

is given by

$$\mathcal{PP} = (\mathcal{G}, \mathbf{w}, \theta_1 \mathbf{f}_1, \dots, \theta_h \mathbf{f}_1, \bar{e}(\mathbf{f}_2, \mathbf{f}_2^*)^{\alpha}),$$
$$\widehat{\mathcal{C}} = \left(C_0 = M_\beta \cdot \bar{e}(\mathbf{f}_2, \mathbf{f}_2^*)^{s\alpha}, \mathbf{c}_1 = s' \mathbf{w} + s'' \mathbf{f}_1 + \mathbb{F}_2 \vec{x}\right)$$

and for $k = 1, \ldots, q$,

$$\begin{aligned} \mathbf{k}_{1}^{(k)} &= \alpha \mathbf{f}_{2}^{*} + r_{1}^{(k)} \mathbb{F}_{1}^{*} \vec{\gamma}^{(k)} + \mathbb{F}_{2}^{*} \vec{y}_{1}^{\prime(k)}, \\ \mathbf{d}_{1,j}^{(k)} &= r_{1}^{(k)} \theta_{j} \mathbf{f}_{j+1}^{*} + \mathbb{F}_{2}^{*} \vec{z}_{j,1}^{\prime(k)} \text{ for } \ell_{k} + 1 \leq j \leq h, \\ \mathbf{k}_{2}^{(k)} &= r_{2}^{(k)} \mathbb{F}_{1}^{*} \vec{\gamma}^{(k)} + \mathbb{F}_{2}^{*} \vec{y}_{2}^{\prime(k)}, \\ \mathbf{d}_{2,j}^{(k)} &= r_{2}^{(k)} \theta_{j} \mathbf{f}_{j+1}^{*} + \mathbb{F}_{2}^{*} \vec{z}_{j,2}^{\prime(k)} \text{ for } \ell_{k} + 1 \leq j \leq h, \end{aligned}$$

where $s' = s - \langle \langle \vec{\chi}_2, \vec{x} \rangle \rangle$, $s'' = s \mathcal{H}_{\hat{\ell}}(\widehat{id}_\beta) - \langle \langle \vec{\chi}_1, \vec{x} \rangle \rangle$ and

$$\begin{split} \vec{y}_1^{\prime(k)} &= \vec{y}_1^{(k)} + r_1^{(k)} \mathbf{Y} \vec{\gamma}^{(k)}, \quad \vec{z}_{j,1}^{\prime(k)} = \vec{z}_{j,1}^{(k)} + r_1^{(k)} \vec{\chi}_2, \\ \vec{y}_2^{\prime(k)} &= \vec{y}_2^{(k)} + r_2^{(k)} \mathbf{Y} \vec{\gamma}^{(k)}, \quad \vec{z}_{j,2}^{\prime(k)} = \vec{z}_{j,2}^{(k)} + r_2^{(k)} \vec{\chi}_2. \end{split}$$

That s' and s'' take the form mentioned above can be seen in the following computations.

$$\begin{aligned} \mathbf{c}_{1} &= s(\mathbf{w} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta})\mathbf{b}_{1}) + \mathbb{B}_{2}\vec{x} \\ &= s(\sum_{j=1}^{\widehat{\ell}}\mathbf{f}_{j+1} + \mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta})\mathbf{f}_{1}) + (\mathbb{F}_{2} - \mathbb{F}_{1}\mathbf{Y})\vec{x} \\ &= \mathbb{F}_{1}(s(\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}), 1, \dots, 1)^{T} - \mathbf{Y}\vec{x}) + \mathbb{F}_{2}\vec{x} \\ &= \mathbb{F}_{1}\left(s\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}), 1, \dots, 1\right)^{T} - (\vec{\chi}_{1}, \vec{\chi}_{2}, \dots, \vec{\chi}_{2})^{T}\vec{x}\right) + \mathbb{F}_{2}\vec{x} \\ &= s\mathbb{F}_{1}(\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}_{\beta}), 1, \dots, 1)^{T} - (\langle\langle\vec{\chi}_{1}, \vec{x}\rangle\rangle, \langle\langle\langle\vec{\chi}_{2}, \vec{x}\rangle\rangle, \dots, \langle\langle\vec{\chi}_{2}, \vec{x}\rangle\rangle)^{T} + \mathbb{F}_{2}\vec{x} \\ &= s'\mathbf{w} + s''\mathbf{f}_{1} + (\vec{x})_{\mathbb{F}_{2}}\end{aligned}$$

The public parameters are the same as in $G_{q,1}$. The additional terms in the semi-functional components of keys are not independent but that does not change the distribution of the sf-components i.e., since $\vec{y}_1^{(k)}, \vec{y}_2^{(k)}, \vec{z}_{1,j}^{(k)}, \vec{y}_{2,j}^{(k)}$ are independent and uniformly distributed, so are $\vec{y}_1^{\prime(k)}, \vec{y}_2^{\prime(k)}, \vec{z}_{1,j}^{\prime(k)}, \vec{y}_{2,j}^{\prime(k)}$. It remains same as in $G_{q,1}$. Let us now take a look at the challenge ciphertext. Ciphertext in our scheme is structured in a way that the coefficient of \mathbf{w} is the real randomiser. Over the basis $(\mathbb{F}, \mathbb{F}^*)$, the coefficient of \mathbf{w} in $\widehat{\mathcal{C}}$ is given by $s' = s - \langle \langle \vec{\chi}_2, \vec{x} \rangle \rangle$. The identity hash is changed to $(s')^{-1}s''$. Unless s' = 0, the hash is a random element of \mathbb{Z}_p due to the choice of $\vec{\chi}_1$. Also C_0 is now given by

$$C_0 = M_\beta \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{s\alpha} = M_\beta \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{\langle\!\langle \vec{\chi}_2, \vec{x} \rangle\!\rangle \alpha} \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{-\langle\!\langle \vec{\chi}_2, \vec{x} \rangle\!\rangle \alpha} \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{s\alpha} = R \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{s'\alpha}$$

where $R = M_{\beta} \cdot \bar{e}(\mathbf{b}_2, \mathbf{b}_2^*)^{\langle\langle \vec{\chi}_2, \vec{x} \rangle\rangle \alpha}$ is a random element of \mathbb{G}_T . Observe that R and the new id-hash are independent since they are determined by $\vec{\chi}_2$ and $\vec{\chi}_1$ (respectively) which are chosen independently at random from \mathbb{Z}_p^{h+1} .

Let F denote the event that s' = 0. Since $\vec{\chi}_2$, \vec{x} and s are all uniformly distributed, we have $\Pr[\mathsf{F}] = 1/p$. $\mathcal{D}_{q,1}$ when expressed over $(\mathbb{F}, \mathbb{F}^*)$ has the same distribution as \mathcal{D}_{final} unless F occurs. Therefore, by difference lemma, we have $|\Pr[X_{q,1}] - \Pr[X_{final}]| \leq \Pr[\mathsf{F}] = \frac{1}{p}$.

5 Conclusion

We have presented a new construction of anonymous HIBE. This is the only known construction fully secure under standard assumptions that can be instantiated with Type-3 pairings. On the other hand, security under standard assumptions is obtained at the cost of working over dual pairing vector spaces of dimension linear in the maximum depth of the HIBE, thus blowing up the size of parameters. It would be interesting to obtain more efficient anonymous HIBE schemes that can be proved secure under standard assumptions.

References

- [ABC⁺05] Michel Abdalla, Mihir Bellare, Dario Catalano, Eike Kiltz, Tadayoshi Kohno, Tanja Lange, John Malone-Lee, Gregory Neven, Pascal Paillier, and Haixia Shi. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 205–222. Springer, 2005.
- [BBG05] Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical Identity-Based Encryption with Constant Size Ciphertext. In Ronald Cramer, editor, EUROCRYPT, volume 3494 of Lecture Notes in Computer Science, pages 440–456. Springer, 2005. Full version available at Cryptology ePrint Archive; Report 2005/015.
- [BGG94] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. Incremental Cryptography: The Case of Hashing and Signing. In Yvo Desmedt, editor, CRYPTO, volume 839 of Lecture Notes in Computer Science, pages 216–233. Springer, 1994.
- [BW06] Xavier Boyen and Brent Waters. Anonymous Hierarchical Identity-Based Encryption (Without Random Oracles). In Cynthia Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2006.
- [CLL⁺12] Jie Chen, Hoon Wei Lim, San Ling, Huaxiong Wang, and Hoeteck Wee. Shorter IBE and Signatures via Asymmetric Pairings. *IACR Cryptology ePrint Archive*, 2012:224, 2012.
- [DCIP10] Angelo De Caro, Vincenzo Iovino, and Giuseppe Persiano. Fully Secure Anonymous HIBE and Secret-Key Anonymous IBE with Short Ciphertexts. In Marc Joye, Atsuko Miyaji, and Akira Otsuka, editors, *Pairing-Based Cryptography - Pairing 2010*, volume 6487 of *Lecture Notes in Computer Science*, pages 347–366. Springer Berlin / Heidelberg, 2010.
- [Duc10] Léo Ducas. Anonymity from Asymmetry: New Constructions for Anonymous HIBE. In Josef Pieprzyk, editor, CT-RSA, volume 5985 of Lecture Notes in Computer Science, pages 148–164. Springer, 2010.
- [LPL13] Kwangsu Lee, JongHwan Park, and DongHoon Lee. Anonymous HIBE with Short Ciphertexts: Full Security in Prime Order Groups. *Designs, Codes and Cryptography*, pages 1–31, 2013.
- [LW10] Allison B. Lewko and Brent Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In Daniele Micciancio, editor, TCC, volume 5978 of Lecture Notes in Computer Science, pages 455–479. Springer, 2010.
- [OT08] Tatsuaki Okamoto and Katsuyuki Takashima. Homomorphic Encryption and Signatures from Vector Decomposition. In Steven D. Galbraith and Kenneth G. Paterson, editors, *Pairing*, volume 5209 of *Lecture Notes in Computer Science*, pages 57–74. Springer, 2008.

- [OT09] Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical Predicate Encryption for Inner-Products. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 214–231. Springer, 2009.
- [PL13] Jong Hwan Park and Dong Hoon Lee. Anonymous HIBE: Compact Construction Over Prime-Order Groups. *IEEE Transactions on Information Theory*, 59(4):2531–2541, 2013.
- [RS12] Somindu C. Ramanna and Palash Sarkar. Anonymous Constant-Size Ciphertext HIBE from Asymmetric Pairings. Cryptology ePrint Archive, Report 2012/057, 2012. http://eprint. iacr.org/.
- [SKOS09] Jae Hong Seo, Tetsutaro Kobayashi, Miyako Ohkubo, and Koutarou Suzuki. Anonymous Hierarchical Identity-Based Encryption with Constant Size Ciphertexts. In Stanislaw Jarecki and Gene Tsudik, editors, Public Key Cryptography, volume 5443 of Lecture Notes in Computer Science, pages 215–234. Springer, 2009.
- [Wat09] Brent Waters. Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions. In Shai Halevi, editor, CRYPTO, volume 5677 of Lecture Notes in Computer Science, pages 619–636. Springer, 2009.

A Hierarchical Identity-Based Encryption

A.1 Definition

A HIBE scheme consists of five probabilistic polynomial time (in the security parameter) algorithms -Setup, Encrypt, KeyGen, Delegate and Decrypt.

- Setup: based on an input security parameter κ , generates and outputs the public parameters \mathcal{PP} and the master secret \mathcal{MSK} .
- KeyGen: inputs an identity vector id and master secret \mathcal{MSK} and outputs the secret key \mathcal{SK}_{id} corresponding to id.
- Encrypt: inputs an identity id, a message M and returns a ciphertext C.
- Delegate: takes as input a depth ℓ identity vector $\mathbf{id} = (\mathbf{id}_1, \dots, \mathbf{id}_\ell)$, a secret key $\mathcal{SK}_{\mathbf{id}}$ and an identity $\mathbf{id}_{\ell+1}$; returns a secret key for the identity vector $(\mathbf{id}_1, \dots, \mathbf{id}_{\ell+1})$.
- Decrypt: inputs a ciphertext C, an identity vector id, secret key \mathcal{SK}_{id} and returns either the corresponding message M or \perp indicating failure.

A.2 Anonymous CPA-Secure HIBE

The following security game captures both anonymity and security against a chosen plaintext attack for a HIBE scheme and will be called ano-ind-cpa.

Setup: The challenger runs the **Setup** algorithm of the HIBE and gives the public parameters to \mathscr{A} .

Phase 1: \mathscr{A} makes a number of key extraction queries adaptively. For a query on an identity vector **id**, the challenger responds with a key \mathcal{SK}_{id} .

Challenge: \mathscr{A} provides two message-identity pairs $(M_0, \widehat{\mathbf{id}}_0)$ and $(M_1, \widehat{\mathbf{id}}_1)$ as challenge with the restriction that neither $\widehat{\mathbf{id}}_0, \widehat{\mathbf{id}}_1$ nor any of their prefixes should have been queried in **Phase 1**. The challenger then chooses a bit β uniformly at random from $\{0, 1\}$ and returns an encryption $\widehat{\mathcal{C}}$ of M_β under the identity $\widehat{\mathbf{id}}_\beta$ to \mathscr{A} .

Phase 2: \mathscr{A} issues more key extraction queries as in **Phase 1** with the restriction that no queried identity id is a prefix of \widehat{id}_{β} .

Guess: \mathscr{A} outputs a bit β' .

If $\beta = \beta'$, then \mathscr{A} wins the game. The advantage of \mathscr{A} in breaking the security of the HIBE scheme in the game ano-ind-cpa given by

$$\mathsf{Adv}_{\mathrm{HIBE}}^{\mathsf{ano-ind-cpa}}(\mathscr{A}) = \left| \Pr[\beta = \beta'] - \frac{1}{2} \right|.$$

The HIBE scheme is said to be (ε, t, q) -ANO-IND-ID-CPA secure if every t-time adversary making at most q queries has $\operatorname{Adv}_{\operatorname{HIBE}}^{\operatorname{ano-ind-cpa}}(\mathscr{A}) \leq \varepsilon$.

B More Assumptions

DDI

Discrete Logarithm in Asymmetric Pairing Groups. The discrete logarithm problem in asymmetric pairings (ADLP) is to compute $x \in \mathbb{Z}_p$ given $(\mathcal{G}, xP_1, xP_2)$. The advantage of an algorithm \mathscr{A} solving ADLP is defined as

$$\mathsf{Adv}^{\mathrm{ADLP}}_{\mathcal{G}}(\mathscr{A}) = \Pr[\mathscr{A}(\mathcal{G}, xP_1, xP_2) = x \,|\, x \xleftarrow{\cup} \mathbb{Z}_p].$$

ADLP is (t, ε) -hard if for every t-time adversary $\mathscr{A} \operatorname{\mathsf{Adv}}_{\mathcal{G}}^{\operatorname{ADLP}}(\mathscr{A}) \leq \varepsilon$.

Decision Diffie-Hellman (DDH). Let $x_1, x_2 \in_U \mathbb{Z}_p$. The DDH problem in \mathbb{G}_1 (DDH1) is to decide, given $(\mathcal{G}, x_1P_1, x_2P_1, Z_1)$, whether $Z_1 = x_1x_2P_1$ or Z_1 is a random element of \mathbb{G}_1 .

Let \mathscr{A} be a probabilistic polynomial time (PPT) algorithm that outputs either 0 or 1. Define its advantage in solving the DDH1 problem as follows.

$$\mathsf{Adv}_{\mathcal{G}}^{\mathrm{DDH1}}(\mathscr{A}) = |\Pr[\mathscr{A}(\mathcal{G}, x_1P_1, x_2P_1, x_1x_2P_1) = 1] - \Pr[\mathscr{A}(\mathcal{G}, x_1P_1, x_2P_1, P_2, Y_1) = 1]|,$$

where $Y_1 \xleftarrow{U} \mathbb{G}_1$. We say that (ε, t) -DDH1 assumption holds in \mathcal{G} if every algorithm \mathscr{A} that runs in time less than or equal to t has $\mathsf{Adv}_{\mathcal{G}}^{\mathsf{DDH1}}(\mathscr{A}) \leq \varepsilon$. Similarly, one can define DDH assumption in \mathbb{G}_2 (DDH2).

C Short IBE of Chen et.al.

A description of the IBE scheme proposed by Chen *et.al.* $[CLL^+12]$ is provided here. We use a compact notation to denote normal and semi-functional ciphertexts and keys. The group elements shown in curly brackets $\{\}$ are the semi-functional components. To get the scheme itself, these components should be ignored.

The description is in terms of DPVS \mathcal{V}_4 over a Type-3 pairing $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P_1, P_2)$ and random dual bases $(\mathbb{B}, \mathbb{B}^*)$ of of \mathcal{V}_4 with $\mathbb{B} = (\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4)$ and $\mathbb{B}^* = (\mathbf{b}_1^*, \mathbf{b}_2^*, \mathbf{b}_3^*, \mathbf{b}_4^*)$. The public parameters, the master secret key, the decryption key and the ciphertext are as follows. α , s and r are uniform random elements of \mathbb{Z}_p .

 $\begin{aligned} \mathcal{PP} &: (\mathbf{b}_1, \mathbf{b}_2, \bar{e}(\mathbf{b}_1, \mathbf{b}_1^*)^{\alpha}). \\ \mathcal{MSK}: \ (\alpha, \mathbf{b}_1^*, \mathbf{b}_2^*). \end{aligned}$

 $C = (C_0 = M \cdot \bar{e}(\mathbf{b}_1, \mathbf{b}_1^*)^{s\alpha}, \ \mathbf{c}_1 = s\mathbf{b}_1 + sid\mathbf{b}_2\{+x_1\mathbf{b}_3 + x_2\mathbf{b}_4\}).$ $S\mathcal{K}_{id} = \alpha\mathbf{b}_1^* + rid\mathbf{b}_1^* - r\mathbf{b}_2^*\{+y_1\mathbf{b}_3^* + y_2\mathbf{b}_4^*\}.$

The randomisers for the semi-functional components are x_1, x_2 and y_1, y_2 in the ciphertext and key respectively.

D Proof of Lemma 4.3

 \mathscr{B} is provided an instance $(\mathcal{G}, xP_1, xP_2)$ of ADLP. It constructs a DPVS \mathcal{V}_{2h+2} over \mathcal{G} and chooses $(\mathbb{B}, \mathbb{B}^*, \psi) \xleftarrow{\mathrm{U}} \mathsf{Dual}(\mathcal{V}_{2h+2})$. The game is simulated as follows.

Setup: \mathscr{B} picks $\alpha, \theta'_1, \ldots, \theta'_h \xleftarrow{U} \mathbb{Z}_p, \delta_1, \ldots, \delta_h \xleftarrow{U} \{0, 1\}$ and implicitly sets

$$\theta_j = \begin{cases} \theta'_j & \text{if } \delta_j = 0\\ x\theta'_j & \text{if } \delta_j = 1 \end{cases}$$

Note that $\theta_j \mathbf{b}_1$ can be computed as $\theta'_j(x\mathbf{b}_1)$ whenever $\delta_j = 1$. Also, since xP_2 is provided in the instance, $\theta_j \mathbf{b}_{j+1}^*$ can be computed. \mathscr{B} can thus generate all public parameters and master secret. It outputs the public parameters and keeps the master secret key.

Key Extraction Phases 1 and 2: \mathscr{A} makes queries on id_1, \ldots, id_q and for id_k , \mathscr{B} returns to \mathscr{A} the output of KeyGen algorithm on id_k .

Challenge: \mathscr{A} sends two message-identity pairs $(M_0, \widehat{\mathbf{id}}_0), (M_1, \widehat{\mathbf{id}}_1)$. Let $\beta \xleftarrow{U} \{0, 1\}$ and $\widehat{\mathbf{id}} = \widehat{\mathbf{id}}_\beta$. Suppose that $\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}) \equiv \mathcal{H}_{\ell_k}(\widehat{\mathbf{id}}_k) \pmod{p}$ for some k. \mathscr{B} aborts the game, outputs x' as the solution to the instance and halts. x' is computed as follows. Let $\mathbf{id}_k = (\mathbf{id}_1, \ldots, \mathbf{id}_\ell), \ \widehat{\mathbf{id}} = (\widehat{\mathbf{id}}, \ldots, \widehat{\mathbf{id}}_{\widehat{\ell}}), \ \widehat{\mathbf{id}}_j = 0$ for $\ell^* + 1 \leq j \leq h$ and $\mathbf{id}_j = 0$ for $\ell + 1 \leq j \leq h$. Define $a = \sum_{\substack{j \in [1,h] \\ \delta_j = 1}} \theta'_j(\widehat{\mathbf{id}}_j - \mathbf{id}_j)$. If $a = 0, \mathscr{B}$ halts with no output. Otherwise, it computes $b = a^{-1} \pmod{p}$ and $x' = b \sum_{\substack{j \in [1,h] \\ \delta_j = 0}} \theta'_j(\mathbf{id}_j - \mathbf{id}_j)$. Let F_k denote the event

that $\mathcal{H}_{\widehat{\ell}}(\widehat{\mathbf{id}}) \equiv \mathcal{H}_{\ell_k}(\mathbf{id}_k) \pmod{p}$.

Claim D.1. $\Pr[a \neq 0] \ge \frac{1}{2}$.

When $a \neq 0$, the x' value computed by \mathscr{B} is a solution for the given instance which the following claim asserts.

Claim D.2. $\Pr[\mathsf{F}_k] \leq 2\mathsf{Adv}^{\mathrm{ADLP}}_{\mathcal{G}}(\mathscr{B})$ for some fixed k.

Proofs for the claims can be found in [BGG94]. Define $\mathsf{F} = \bigcup_{k=1}^{q} \mathsf{F}_{k}$. The two games G_{actual} and $\mathsf{G}_{restricted}$ proceed the same way unless F occurs. By the difference lemma, we have $|\Pr[X_{actual}] - \Pr[X_{restricted}]| \leq \Pr[\mathsf{F}] \leq \sum_{k=1}^{q} \Pr[\mathsf{F}_{k}]$ (by the union bound). Therefore, $|\Pr[X_{actual}] - \Pr[X_{restricted}]| \leq 2q\mathsf{Adv}_{\mathcal{G}}^{\mathrm{ADLP}}(\mathscr{B})$.